ACCESS.bus

by Michael Burton

A New Peripheral Bus

Would you like to be able to plug a keyboard, two mice, a trackball, a modem and a printer into a single port on your PC? And then, while the computer is still powered up, unplug one of the mice and plug in a bar code reader? You can do all of this and more with ACCESS.bus.

Digital Equipment Corporation and Philips/Signetics have joined forces to propose this new open desktop connectivity standard. The ACCESS.bus Industry Group (ABIG) has been formed to regulate and promote the new bus. ABIG members include DEC, Honeywell, Logitech, Philips and Sun Microsystems, to name just a few.

There are many advantages to the ACCESS.bus standard. It is low cost, dynamically reconfigurable, relatively inexpensive and the interface is uniform for all devices. ACCESS.bus is also an open standard, unlike Apple Computer's comparable Apple Data Bus (ADB).

Bus Description

ACCESS.bus allows multiple peripheral devices to be simultaneously supported on a single computer port in a daisy chain fashion, somewhat like a SCSI daisy chain. These peripherals may operate simultaneously at transfer rates of up to 125,000 baud, with a maximum data throughput rate of approximately 80,000 baud. This is ideal for low speed peripherals such as keyboards, modems, trackballs and mice.

A maximum cable length of 8 meters is permitted. A four pin, shielded rectangular connector is used on ACCESS.bus cables. The maximum amount of power available is 1 amp at 5 volts. The bus can support up to 125 peripheral devices, but the normal practical limit is 14. By way of comparison, the Apple Data Bus supports 5 meters of cable and 3 devices with a data rate of 10,000 baud.

ACCESS.bus is a layered protocol. There are three layers; the hardware protocol layer, the Base protocol layer and the Application protocol layer. Using a postal analogy, the hardware protocol layer is the mail carrier, the Base protocol layer is the envelope and the Application protocol layer is the contents of the envelope.

The hardware protocol layer is based on the I^2C (Inter-Integrated Circuit) serial protocol, which is directly supported by the Philips/Signetics 8051 family of microprocessors (80CL410, 80C552, 80C652, 80C528, 87C654 and 87C751). This protocol defines a scheme for performing bus transactions, including message addressing, the framing of bits into bytes and the acknowledgment of each byte by the receiver.

The Base protocol layer is a software protocol that is common to all ACCESS.bus devices and that builds on the hardware protocol layer to establish the connection between the computer and a number of peripheral devices. The Base protocol layer specifies device power-up, identification, addressing and the message envelope for device-specific data and control information.

The Application protocol layer is a software protocol that differentiates message contents for specific kinds and classes of devices.

Hardware Protocol Layer

The hardware I^2C protocol layer is a 2-wire (clock and data) serial protocol that allows wire-AND connection of peripheral devices to the clock and data lines. Any device may be either a master (controlling the transaction and generating the clock) or a slave for any given bus transaction. Several masters can contend for the bus and an arbitration scheme resolves bus mastership without data loss or retransmission. The clock rates for various peripherals may vary widely, since the bus has a cooperative serial clock synchronization scheme.

Base Protocol Layer

The Base protocol layer contains definitions for a number of control and status messages that are common to all ACCESS.bus peripherals. The messages are used for the configuration process, where peripherals are recognized, assigned unique address identifiers and are then connected with appropriate device drivers to enable an application program to talk to them. A message has five parts to it:

1.)The first byte is the address of the destination or receiver.

2.)The second byte is the address of the source or transmitter.

3.)The third byte specifies whether the body of the message is control or data, if there are any sub-devices (0 to 3) and the length of the message body.

4.)The part is the message body, which can be from 0 to 127 bytes in length.

5.)The last byte is the checksum, a bit-wise exclusive-or of all the preceding bytes in the message.

There are eight base messages, shown below

Computer to Device Messages

1.)Reset - Force the device to its power-up state and to its default I^2C address.

2.)IdRequest - Ask the device for its identification string.

3.)AssignAddress - Tell the device with a matching identification string to change its address to a new address.

4.)CapRequest - Ask the device to send its capabilities information.

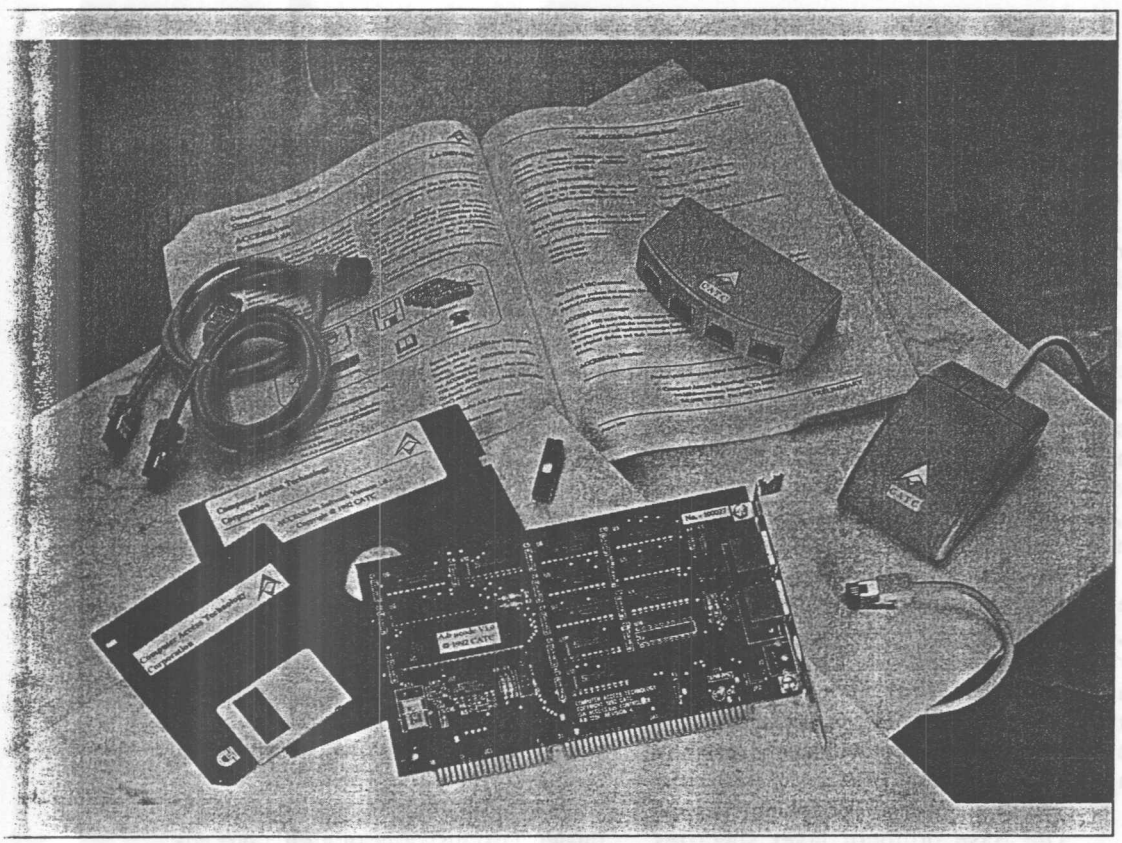
Device to Computer Messages

1.)Attention - Inform the computer that the device has finished its power-up/reset tests and that it needs to be configured.

2.)IdReply - Reply to an IdRequest with the device's unique identification string.

3.)CapReply - Reply to a CapRequest with a fragment of the device's capabilities string.

4.)IfError - Invalid checksum or premature end of message detected.



ACESS.bus Development Kit

When a peripheral device powers up or resets, its initial device address is always 6Eh. The Base messages are used to reset this device address to a unique address between 02h and 7Eh (125 assignable addresses).

Application Protocol Layer

Application protocol layer messages are specific to the peripheral device and the message layouts are different for each device type, as well as for each device sub-type. This means that the device drivers are different at this level, but since the hardware and Base protocols are device-independent, much of the firmware support code can be shared by different devices. Even at the Applications protocol level, a common message structure is used for similar device types, so that a common approach can be used in writing ACCESS.bus device drivers. To date, peripheral devices for ACCESS.bus have been classed into three broad categories:

Keyboards - May have as many as 255 keys. Special function keys and annunciators are supported.

Locator devices - Includes pointing devices such as mice, trackballs, graphic tablets, etc. Provides for devices with up to 15 degrees of freedom and up to 16 binary keys.

Text devices - Devices that support data streams (e.g., modems, printers, etc.).

What ACCESS.bus Means to You

ACCESS.bus is a coherent, well-defined desktop connectivity standard. It is solidly backed by DEC, Philips/Signetics and others in the industry and it offers much to both users and peripheral manufacturers. Since it is so new, there are plenty of opportunities for entrepreneurs to implement ACCESS.bus devices with fewer direct challenges from the big guys. Peripheral devices that already use an 8051 microprocessor only need to change their I/O firmware, so the impact of designing for a new bus can be minimized. Watch out, though—Microsoft and others are starting to pay attention to ACCESS.bus, so the window of opportunity is getting smaller.

ACCESS.bus Access

The ACCESS.bus Industry Group will provide free information, including the ACCESS.bus Specification, to anyone who asks for it. It costs nothing to join ABIG (as a company) if you plan to develop an ACCESS.bus peripheral. ABIG's address is:

ACCESS.bus Industry Group
370 Altair Way Suite 215
Sunnyvale, CA 94086
408-991-3517 or FAX:408-991-3773

At least one company provides an ACCESS.bus Development Kit, at a cost of $1500. The kit includes a controller board, a Logitech ACCESS.bus mouse, an expansion box, two cables, a Philips/Signetics 87C751 microprocessor, a comprehensive software package and documentation for everything. Their address is:

Computer Access Technology Corporation (CATC)
3375 Scott Boulevard, Suite 410
Santa Clara, CA 95054
800-909-2282 or 408-727-6600

ME

Michael Burton is a Senior Software Engineer, at Key Tronic Corporation, Spokane, WA

This article is reprinted with permission from the May/June 1993 issue of ***Midnight Engineering***, an entrepreneurial engineering magazine. To obtain a sample issue, call 719-254-4558 or fax:719-254-4517.